IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Michael Swafford et al.	Examiner:	Jared Ian Rutz
Application No.:	10/750,455	Group Art Unit:	2187
Filed:	December 31, 2003	Docket No.:	50037.0237US01
Title:	OVERWRITE DETECTION DIAGNOSTIC FOR MEMORY HEAP		

Electronically filed February 20, 2008

AMENDMENT

Mail Stop AF
Commissioner for Patents
Attn: Examiner Jared Ian Rutz
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Final Office Action of August 20, 2007, please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of **Claims** which begins on page 2 of this document.

Remarks, including a summary of changes to the claims, begin on page 9 of this document.

CLAIMS

1. (Currently Amended) A method for providing overwrite detection for an allocable memory block, comprising:

receiving a request for performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

performing a checksum on the allocable memory block;

storing results of the checksum within the allocable memory block;

generating an overwrite detection pattern for the allocable memory block based on a size of an area of the allocable memory block that is used for alignment purposes, wherein

the area includes a number of memory units each including a plurality of bytes;

a predetermined code is generated for one or more of the bytes in each of the memory units so that a change in the predetermined numeric code in one or more of the bytes of one or more of the memory units indicates a memory overwrite; and

the predetermined code generated for the one or more bytes in each of the one or more memory units is different for each of the memory units;

storing the overwrite detection pattern in the area of the allocable memory block used for alignment purposes, wherein the overwrite detection pattern is stored separately from the results of the checksum in the allocable memory block;

checking the overwrite detection pattern; and

forcing an access violation if one of the checksum is not valid and the overwrite detection pattern has been modified.

2. (Previously Presented) The method of Claim 1, further comprising examining a heap to determine whether the overwrite detection pattern has been overwritten.

3. (Canceled).

4. (Previously Presented) The method of Claim 1, further comprising examining the results of the checksum to determine the presence of memory errors.

5. (Previously Presented) The method of Claim 1, wherein the overwrite detection pattern is written at an end of the allocable memory block opposite another end of the allocable memory block where the results of the checksum are stored.

6. (Original) The method of Claim 1, wherein a logical function of the elements within the overwrite detection pattern provides a predetermined result.

7. - 8. (Canceled).

9. (Original) The method of Claim 1, further comprising storing a heap index for the allocable memory block within the allocable memory block, wherein the heap index points to one of a plurality of heaps.

10. (Previously Presented) The method of Claim 1, further comprising storing a timestamp within the allocable memory block, wherein the timestamp indicates the time when requesting the allocable memory block is performed.

11. (Currently Amended) A computer storage medium storing computer readable instructions for overwrite detection within an allocable memory block, comprising:

a first component that is arranged to receive a request for performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

a second component that is arranged to generate an overwrite detection pattern for the allocable memory block wherein the overwrite detection pattern is written at an end of the allocable memory block opposite another end of the allocable memory block in which a header for the allocable memory block is stored and the overwrite detection pattern is based on a size of an area of the allocable memory block that is used for alignment purposes, wherein

the area includes a number of memory units each including a plurality of bytes;

a predetermined code is generated for one or more of the bytes in each of the memory units so that a change in the predetermined numeric code in one or more of the bytes of one or more of the memory units indicates a memory overwrite; and

the predetermined code generated for the one or more bytes in each of the one or more memory units is different for each of the memory units;

a third component that is arranged to store the overwrite detection pattern in the area of the allocable memory block used for alignment purposes in the allocable memory block;

a fourth component that is arranged to generate a checksum on the allocable memory block;

a fifth component that is arranged to store results of the checksum in the header of the allocable memory block; and

a sixth component that is arranged to store a heap index for the allocable memory block within the allocable memory block, wherein the heap index points to one of a plurality of heaps.

12. (Previously Presented) The computer storage medium of Claim 11, further comprising an examination component that is arranged to examine one of the plurality of heaps heap to determine whether the overwrite detection pattern has been overwritten.

13. (Canceled).

14. (Previously Presented) The computer storage medium of Claim 13, further comprising a checksum examination component that is arranged to examine results of the checksum to determine the presence of memory errors.

15. (Previously Presented) The computer storage medium of Claim 11, wherein the overwrite detection pattern is written at an end of the allocable memory block opposite another end of the allocable memory block where the results of the checksum are stored.

16. (Previously Presented) The computer storage medium of Claim 11, wherein a logical function of the elements within the overwrite detection pattern provides a predetermined result.

17. (Canceled).

18. (Previously Presented) The computer storage medium of Claim 11, wherein the overwrite detection pattern is checked and an access violation is forced if the overwrite detection pattern has been modified.

19. (Canceled).

20. (Previously Presented) The computer storage medium of Claim 11, further comprising a timestamp component that is arranged to store a timestamp within the allocable memory block, wherein the timestamp indicates the time when requesting the allocable memory block is performed.

21. (Currently Amended) A system for overwrite detection in an allocable memory block, comprising:

a computer memory that comprises a heap in which an allocable memory block can be allocated and freed;

a memory allocator that is arranged to receive a request for performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;

a pattern generator that is arranged to generate an overwrite detection pattern for the allocable memory block based on a size of an area of the allocable memory block that is used for alignment purposes, wherein

the area includes a number of memory units each including a plurality of bytes;

a predetermined code is generated for one or more of the bytes in each of the memory units so that a change in the predetermined numeric code in one or more of the bytes of one or more of the memory units indicates a memory overwrite; and

the predetermined code generated for the one or more bytes in each of the one or more memory units is different for each of the memory units;

an allocable memory block formatter that is arranged to store the overwrite detection pattern in the allocable memory block; and

a memory timestamp system that is arranged to store a timestamp within the allocable memory block, wherein the timestamp indicates one of the time of[[:]] one of the requesting of the allocable memory block ~~once the allocable memory block has been allocated;~~ and the freeing of the allocable memory block ~~once the memory block has been freed~~.

22. (Previously Presented) The system of Claim 21, further comprising a memory verification system that is arranged to examine a heap to determine whether the overwrite detection pattern has been overwritten.

23. (Previously Presented) The system of Claim 21, further comprising a memory verification system that is arranged to perform a checksum on the allocable memory block and storing results of the checksum within the allocable memory block.

24. (Original) The system of Claim 23, further comprising a memory verification system that is arranged to examine the results of the checksum to determine the presence of memory errors.

25. (Previously Presented) The system of Claim 23, wherein the overwrite detection pattern is written at an end of the allocable memory block opposite another end of the allocable memory block where the results of the checksum are stored.

26. (Original) The system of Claim 21, wherein a logical function of the elements within the overwrite detection pattern provides a predetermined result.

27. (Canceled).

28. (Original) The system of Claim 21, wherein the overwrite detection pattern is checked and an access violation is forced if the overwrite detection pattern has been modified.

29. (Original) The system of Claim 21, further comprising a memory indexing system that is arranged to store a heap index for the allocable memory block within the allocable memory block, wherein the heap index points to one of a plurality of heaps.

30. (Canceled).

31. (Previously Presented) The method of Claim 1, wherein the overwrite detection pattern is checked when the allocable memory block is passed back to the operating system.

32. (Previously Presented) The computer storage medium of Claim 18, wherein the overwrite detection pattern is checked when the allocable memory block is passed back to the operating system.

33. (Previously Presented) The system of Claim 28, wherein the overwrite detection pattern is checked when the allocable memory block is passed back to the operating system.

34. (Previously Presented) The method of Claim 1, further comprising storing a timestamp within the allocable memory block, wherein the timestamp indicates the time when freeing the allocable memory block is performed.

35. (Previously Presented) The computer storage medium of Claim 11, further comprising a timestamp component that is arranged to store a timestamp within the allocable memory block, wherein the timestamp indicates the time when freeing the allocable memory block is performed.

REMARKS

Claims 1, 2, 4-7, 9-12, 14-18, 20-29, and 31-35 were pending at the time the Office Action was issued, of which claims 1, 11, and 21 are independent claims, with claims 3, 8, 13, 19, and 30 having been previously canceled.

Claims 7, 17, and 27 are presently canceled.

Claims 1, 11, and 21 are currently amended.

Thus, claims 1, 2, 4-6, 9-10, 12, 14-18, 20-26, 28-29, and 31-35 remain pending.

Applicants' representative is very grateful to the Examiner for contacting him to discuss the status of the case. Thank you very much.

Claim Rejections under 35 U.S.C. § 112

Applicants have amended claim 21 to correspond with the permitted amendment to the specification to recite subject matter included in the originally presented claims. Applicants submit that the amendment resolves the rejection under 35 U.S.C. § 112 with regard to claim 21 and claims 22-26 and 28-29 that depend from claim 21.

Claim Rejections under 35 U.S.C. § 102

Claims 1-2, 4-7, 9, 11-12, 14-18, and 31-32 were rejected under 35 U.S.C. § 102(e) as having been anticipated by U.S. Patent No. 7,181,585 to Abrashkevich et al. (hereinafter "Abrashkevich"). The cancellation of claims 7 and 17 renders moot the rejection to those

claims. Moreover, applicants respectfully submit that amendments to independent claims 1 and 11 renders moot the rejections to those claims as well as claims depending from claims 1 and 11.

Claim 1 as amended recites the generation of an overwrite pattern that is neither taught nor suggested by Abrashkevich or any other references. Claim 1 is reproduced below for the convenience of the Examiner:

> 1. (Currently Amended) A method for providing overwrite detection for an allocable memory block, comprising:
> receiving a request for performing one of requesting the allocable memory block, requesting the size of the allocable memory block, and freeing the allocable memory block;
> performing a checksum on the allocable memory block;
> storing results of the checksum within the allocable memory block;
> generating an overwrite detection pattern for the allocable memory block based on a size of an area of the allocable memory block that is used for alignment purposes, wherein
> the area includes a number of memory units each including a plurality of bytes;
> a predetermined code is generated for one or more of the bytes in each of the memory units so that a change in the predetermined numeric code in one or more of the bytes of one or more of the memory units indicates a memory overwrite; and
> the predetermined code generated for the one or more bytes in each of the one or more memory units is different for each of the memory units;
> storing the overwrite detection pattern in the area of the allocable memory block used for alignment purposes, wherein the overwrite detection pattern is stored separately from the results of the checksum in the allocable memory block;
> checking the overwrite detection pattern; and
> forcing an access violation if one of the checksum is not valid and the overwrite detection pattern has been modified.

Neither Abrashkevich nor any other references teach or suggest the limitations recited by claim 1 regarding the generation of an overwrite detection pattern. For example, although Abrashkevich describes using "a well-known signature (eye catcher) which is used to verify memory" (Abrashkevich, Column 19, Lines 28-30, cited by the Office Action in Paragraph 9.b.),

this general suggestion of a signature does not teach what is suggested by this limitation for at least two reasons. First, as noted by the Office Action in Paragraph 9.b., Abrashkevich teaches using a signature that fits in a "reserved 32 bit area." (Abrashkevich, Column 19, Line 28). Claim 1 recites an overwrite pattern that is not limited to a 32 bit area or an area of any fixed size but, instead, fits the area used for alignment of the allocable block. Second, Abrashkevich describes "*a* well-known signature," which teaches using a singular signature recorded in the 32 bit reserved area. (Abrashkevich, Column 19, Lines 28-29; emphasis added). By contrast, claim 1 recites an overwrite pattern in which the overwrite pattern "a predetermined code generated for the one or more bytes in each of the one or more memory units" that is "is different for each of the memory units." Abrashkevich fails to teach these limitations thus the rejection under 35 U.S.C. § 102(e) respectfully must be withdrawn against claim 1.

Applicants submit that claim 1 is in condition for allowance. Moreover, claims 2, 4-6, 9, and 34 depend from and apply additional limitations to claim 1. Claims 2, 4-6, 9, and 34 are thus allowable for at least the same reasons for which claim 1 is allowable. Thus, applicants submit that claims 1-2, 4-7, 9, and 34 all are in condition for allowance.

Applicants submit similar amendments to independent claim 11. For the same reasons, applicants respectfully submit that claim 11 as well as claims 12, 14, 18, and 35 depending from claim 11, also all are in condition for allowance.

Furthermore, applicants submit similar amendments to independent claim 21. Thus, for the same reasons, applicants respectfully submit that claim 21 as well as claims 22-26, 29, and 31-33 also are in condition for allowance.

Claim Rejections under 35 U.S.C. § 103

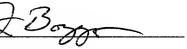
Claims 10, 20-29, and 33-35 rejected under 35 U.S.C. § 103(a) as being unpatentable over Abrashkevich in view of other references. Respectfully, because of the amendments to claims 1, 11, and 21 and/or the cancellation of claims 8, 17, and 27, the rejections under 35 U.S.C. § 103(a) are rendered moot. Applicants thus respectfully request that the rejections under 35 U.S.C. § 103(a) be withdrawn and that each of these claims be found in condition for allowance.

CONCLUSION

In view of the foregoing amendments and remarks, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicants at the telephone number provided below.

Respectfully submitted,

MERCHANT & GOULD P.C.



Frank J. Bozzo
Registration No. 36,756
Direct Dial: 206.342.6294

MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, Minnesota 55402-0903
206.342.6200